

H I G H - R E S O L U T I O N
C O L O R G R A P H I C S
O N T H E
A P P L E - I I C O M P U T E R

S. Wozniak (WOZ)

APPLE COMPUTER, INC.

November 30, 1977

APPLE-II HI-RES GRAPHICS SUBROUTINES

The APPLE-II computer comes with a high-resolution (hereafter 'HI-RES') color graphics display mode of 280 horizontal by 192 vertical resolution. Because 8K bytes of RAM are dedicated solely to maintaining the HI-RES display, a minimum 12K byte system (configured for HI-RES) is required to use this mode. For practical reasons, 16K bytes is the strongly recommended minimum. A 6502 machine language subroutine package has been developed to simplify efficient use of the APPLE-II HI-RES display for assembly language and BASIC programmers. The routines for initializing the HI-RES display, plotting points, drawing lines, and drawing shapes are described herein.

USING THE HI-RES SUBROUTINES

Despite the fact that HI-RES graphics commands are not built into APPLE-II BASIC, a convenient scheme for referencing the subroutines and their parameters by name has been devised, as illustrated below.

TRADITIONAL METHOD OF CALLING MACHINE LANGUAGE SUBROUTINES

```
> POKE 800, X MOD 256
> POKE 801, X / 256
> POKE 802, Y
> POKE 812, C    (color)
> CALL 2834
```

IMPROVED METHOD

```
> X0 = X
> Y0 = Y
> COLR = C
> CALL PLOT
```

The first statement of a program using the HI-RES subroutines should be as follows

```
0    X0 = Y0 = COLR = SHAPE = ROT = SCALE
```

The purposes of this statement are to define a line number 0 (necessary when later appending the HI-RES PREFIX program) and to enter the first 6 BASIC variable names in the symbol table in a fixed sequence. When executed, each of the 6 parameters will be assigned storage at fixed locations relative to the address contained in the BASIC 'start of variables' pointer, LOWEM, making them readily accessible by the HI-RES subroutines.

Different parameter names may be used provided that they retain the same number of characters. This is necessary to insure that the storage locations for each relative to LOMEM do not change. For example, the name XX could be used in place of X0 but XCOORD could not.

The parameters SHAPE, ROT, and SCALE are used only by the HI-RES shape draw subroutines and may be omitted from programs using only the PLOT and LINE features. Omitting unnecessary variable definitions is one method of enhancing the overall performance of some BASIC programs on the APPLE-II and is thus desirable.

FIRST LINE OF PROGRAMS NOT USING THE SHAPE DRAW SUBROUTINES

```
0  X0 = Y0 = COLR
```

After the parameter names have been defined, the HI-RES subroutine names themselves may be defined and assigned corresponding subroutine entry addresses as values. Calling subroutines by name is preferable to calling them by entry address because the entry addresses may vary in future versions of the HI-RES subroutines, and names are better self documenting.

Absolute CALLCALL by name

	5	INIT = 2048			
	.		.		
	.		.		
	.		.		
100	CALL	2048	100	CALL	INIT
	.			.	
	.			.	
	.			.	
200	CALL	2048	200	CALL	INIT

In the above CALL by name example, should the INIT subroutine entry address change to -12288, only line 5 need be changed. In the absolute CALL example, lines 100 and 200 (and any others referencing the INIT subroutine) will have to be changed. The self documenting advantage of the CALL by name example should be apparent.

The following statement lists all HI-RES subroutine entry initializations available to BASIC programs. Other names may be used at the programmer's discretion.

```
5  INIT = 2048 : CLEAR = 2062 : BKGND = 2865 :  
   POSN = 2809 : PLOT = 2830 : LINE = 2836 :  
   DRAW = 2871 : DRAW1 = 2874 : XDRAW = 2884 :  
   XDRAW1 = 2887 : FIND = 2556
```

The allowable color specification values may also be referenced by name, if the initialization statement below is included in your program. Note that 'GREEN' is preceded by 'LET' to avoid a syntax error due to confusion with the GR command.

```
7  BLACK = 0 : LET GREEN = 42 : VIOLET = 85 : WHITE = 127
```

If your APPLE-II has been modified for additional HI-RES colors, the following assignments are also valid.

```
8  ORANGE = 170 : BLUE = 213 : BLACK2 = 128 : WHITE2 = 255
```

Unnecessary variable definitions should be avoided as they will slow some programs. Therefore, a program should not define VIOLET = 85 unless it uses the color VIOLET. The example below illustrates condensed initialization statements for a program using only the INIT, PLOT, and DRAW subroutines, and the colors GREEN and WHITE.

```
0  XO = YO = COLR = SHAPE = ROT = SCALE
5  INIT = 2048 : PLOT = 2830 : DRAW = 2871
7  LET GREEN = 42 : WHITE = 127
```

.
.
.
.

In extreme cases any of the following techniques will further enhance program performance.

- (1) Omit the color and subroutine name initializations.

Refer to colors and subroutines by value, not name.

This does not apply to the parameter references.

- (2) Define the most frequently used program variable names prior to the subroutine name and color name initializations (lines 5 and 7 in the prior examples). The example below will speed up programs extensively referencing variables I, J, and K.

```
0  X0 = Y0 = COLR = SHAPE = ROT = SCALE
```

```
2  I = J = K
```

```
5  INIT = 2048 : CLEAR = 2062 : BKGND = 2865 :
```

```
    POSN = 2809 ..... etc.
```

```
7  BLACK = 0 : LET GREEN = 42 : ..... etc.
```

- (3) Use the parameter names as program variables when possible. Because they are defined first, the parameters are the most quickly accessed BASIC variables.

INITIALIZATION SUBROUTINES

The normal HI-RES display consists of a 280 horizontal by 160 vertical grid above 4 lines of text and is initiated with the BASIC command below.

> CALL INIT

The INIT subroutine also clears the HI-RES display and initializes other HI-RES subroutines. After calling INIT the programmer may eliminate the 4 line text display, extending the HI-RES display to a 192 vertical resolution, with the following command:

> POKE -16302,0

The 4-line text display may be restored at any time as follows:

> POKE -16301,0

Valid X-coordinates vary from 0 (leftmost) to 279 (rightmost)
Valid Y-coordinates vary from 0 (topmost) to 159 or 191 (bottommost)
depending on whether or not the 4 line text display is enabled.

At any time after INIT has been called, the entire HI-RES display may be cleared with the CLEAR subroutine as shown below.

> CALL CLEAR

The HI-RES display may be quickly set to any background color with the BKGND subroutine. BKGND expects a color specification in the BASIC variable COLR. The example below turns the entire HI-RES display green.

```
0  XO = YO = COLR
5  INIT = 2048 : BKGND = 2865 :
   LET GREEN = 42
10 CALL INIT
20 COLR = GREEN
30 CALL BKGND
40 END
```

Only the colors previously mentioned (BLACK, GREEN, VIOLET, and WHITE) may be specified in COLR. Do not make up your own. For example, COLR = YELLOW is not allowed.

If COLR is greater than 255 when BKGND is called then a range error will occur. The message '(beep) *** RANGE ERR' will be displayed and the program will halt.

POINTS AND LINES

The PLOT subroutine is used to plot a single point of the HI-RES display in a specified color. COLR must be less than 255, X0 must be 0 to 279, and Y0 must be 0 to 191 when PLOT is called or a range error will result and the program will halt. The program below plots one white dot at X-coordinate 35, Y-coordinate 55 (35,55) and one at (85,90).

```
0  X0 = Y0 = COLR
5  INIT = 2048 : PLOT = 2380 : WHITE = 127
10 CALL INIT
20 COLR = WHITE
30 X0 = 35 : Y0 = 55 : CALL PLOT
40 X0 = 85 : Y0 = 90 : CALL PLOT
50 END
```

Connecting any two coordinates with a straight line is almost as easy as plotting points. After plotting one endpoint as shown in the example above, the other endpoint is specified in X0 and Y0 and the the LINE subroutine is called. As with the PLOT subroutine, COLR must be less than 256, X0 must be 0 to 279, and Y0 must be 0 to 191 or a range error will result and the program will halt. The following example draws a white line from (35,40) to (170,100), a green line from (270,10) to (5,145), and a violet line from (20,70) to (190,110).

```
0  XO = YO = COLR
5  INIT = 2048 : PLOT = 2830 : LINE = 2836 :
   LET GREEN = 42 : VIOLET = 85 : WHITE = 127
10 CALL INIT
20 COLR = WHITE : XO = 35 : YO = 40 : CALL PLOT
25 XO = 170 : YO = 100 : CALL LINE
30 COLR = GREEN : XO = 270 : YO = 10 : CALL PLOT
35 XO = 5 : YO = 145 : CALL LINE
40 COLR = VIOLET : XO = 20 : YO = 70 : CALL PLOT
45 XO = 190 : YO = 110 : CALL LINE
50 END
```

The following example illustrates that the parameter variables may be used as FOR loop indices. Horizontal violet lines are drawn on a green background at every tenth vertical coordinate.

```
0  XO = YO = COLR
5  INIT = 2048 : BKGND = 2865 : PLOT = 2830 :
   LINE = 2836 : LET GREEN = 42 : VIOLET = 85
10 CALL INIT
20 COLR = GREEN : CALL BKGND
30 COLR = VIOLET
40 FOR YO = 5 TO 155 STEP 10
50 XO = 10 : CALL PLOT : XO = 270 : CALL LINE
60 NEXT YO : END
```

Multiple lines which are connected endpoint to endpoint may be drawn without intervening PLOT calls. In the example below, a white line connects (10,20) to (250,70), and green line connects (250,70) to (20,150), and a violet line connects (20,150) to (260,30).

```
0  XO = YO = COLR
5  INIT = 2048 : PLOT = 2830 : LINE = 2836 :
   LET GREEN = 42 : VIOLET = 85 : WHITE = 127
10 CALL INIT
20 COLR = WHITE : XO = 10 : YO = 20 : CALL PLOT
30 XO = 250 : YO = 70 : CALL LINE
40 XO = 20 : YO = 150 : COLR = GREEN : CALL LINE
50 XO = 260 : YO = 30 : COLR = VIOLET : CALL LINE
60 END
```

CAUTION

Do not attempt to draw a line prior to the first PLOT. Because the first endpoint has not been defined, the line may be drawn in random memory locations, not necessarily restricted to the screen memory.

DRAWING SHAPES

Up to 255 different shapes may be defined, edited, and saved on a single tape

After loading the HI-RES subroutines such a 'shape tape' (containing a 'shape table') may be read as follows.

1. Position shape tape in recorder.
2. Load shape tape with the BASIC command:

> CALL 3001
3. Start recorder (PLAY).
The above command immediately begins reading tape.
4. Wait for two beeps.

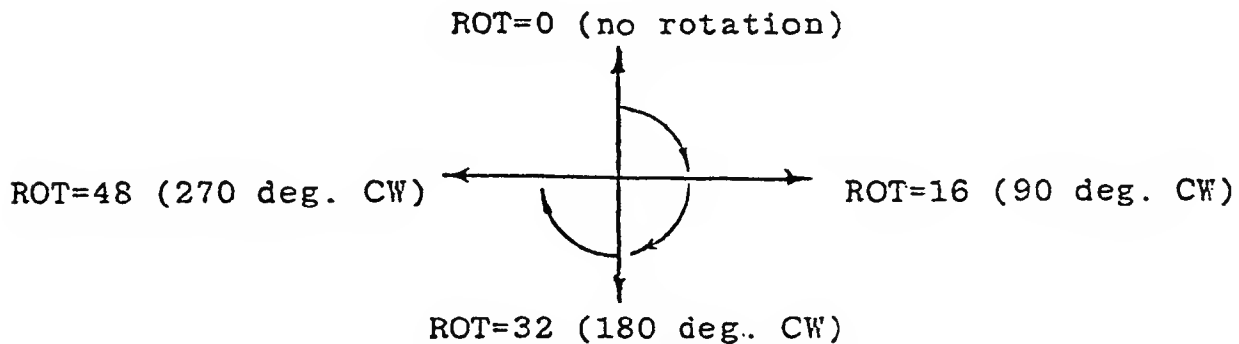
Shape tables always load beginning at address \$C00 with the HI-RES subroutines in locations \$800-\$BFF. Upon loading a shape table, the BASIC 'start of variables' pointer LOMEM is set to contain the address of the location immediately following the last shape table byte.

If not enough free memory is available to contain the shape table then the message '(beep) *** MEM FULL ERR' will be displayed. If no beep is heard when loading a shape tape then something is probably wrong with the tape connection and you will have to hit RESET and type C^C (Control-C) to reenter BASIC. If you hear a single beep and then the system hangs it means your shape tape is probably bad and after hitting RESET and typing C^C you may have to restore the LOMEM setting to SC00 (3072) as follows.

> LOMEM : 3072

The DRAW subroutine is used to display any of the predefined shapes included in the current shape table. The origin or 'beginning point' of the shape is specified in X0 and Y0 and the color is specified in COLR as with PLOT. The shape number desired is specified in SHAPE. For example, SHAPE = 3 specifies that the third shape of the current shape table is to be drawn. A scale factor is specified in the variable SCALE and a rotation in ROT. A scale factor of 4 implies a shape 4 times the defined size. A scale factor of 0 is always interpreted as 256.

Rotations



COLR must be 0 to 255, X0 must be 0 to 279, Y0 must be 0 to 191, ROT must be 0 to 255 (due to MOD 64 arithmetic, ROT=64 is equivalent to ROT=0), SCALE must be 0 to 255, and SHAPE must be greater than zero and less than or equal to the current number of shape table shapes or else a range error will result when DRAW is called and the program will halt. In other words, the programmer will always be notified if HI-RES subroutines are called with any invalid parameters.

The following program example draws shape number 3 in white at a 90 degree clockwise rotation and scale factor of 2. The origin is at (140,80). It is assumed that a shape table with at least 3 shape definitions has been loaded.

```
0  XO = YO = COLR = SHAPE = ROT = SCALE
5  INIT = 2048 : DRAW = 2871
7  WHITE = 127
10 CALL INIT
20 XO = 140 : YO = 80 : COLR = WHITE
30 SHAPE = 3 : ROT = 16 : SCALE = 2
40 CALL DRAW
50 END
```

The XDRAW subroutine is identical in operation to the DRAW subroutine except that the defined shape is exclusive-ored (EX-OR'd) onto the screen. The EX-OR operation complements all screen memory bits of the shape, 0's become 1's and vice-versa. No color need be specified. A unique property of XDRAW is that 2 successive calls with identical parameters will first cause a shape to be drawn (in white) and then erased. The following program example causes the rotation of shape number 3 to track paddle 0. XDRAW is used for both the draw and erase operations. Although the color is not optional, the variable COLR may not be omitted from the parameter declarations (line 0) or the SHAPE, ROT, and SCALE parameters will not be assigned storage in their standard locations relative to LOMEM.

```

0  XO = YO = COLR = SHAPE = ROT = SCALE
5  INIT = 2048 : XDRAW = 2884
10 CALL INIT
20 XO = 140 : YO = 80 : SHAPE = 3 : SCALE = 2
30 R = 0 : GOTO 60 : REM  DRAW FIRST SHAPE
40 R = PDL(0) : IF R = ROT THEN GOTO 30
50 CALL XDRAW : REM  ERASE AT OLD ROT
60 ROT = R : CALL XDRAW : REM  DRAW AT NEW ROT
70 GOTO 40 : REM  CHECK FOR ROT CHANGE
80 END

```

DRAW1 and XDRAW1 are identical to DRAW and XDRAW respectively except that the most recently plotted (or drawn) point serves as the shape origin and the current color is not updated. Thus XO, YO, and COLR are not specified.

If you draw a shape and then wish to draw a line from the final plot position of that shape to a fixed coordinate, you may do so. After drawing the shape, however, you must call FIND prior to calling LINE. The FIND subroutine determines the X-Y coordinates of the final shape plot position (or current plot position if used after other subroutines) and uses it as the beginning endpoint of the following call to LINE. The following program example draws a shape and then a violet line from the final plot position of the shape to (10,25).


```

0  XO = YO = COLR = SHAPE = ROT = SCALE
5  INIT = 2048 : LINE = 2836 : DRAW = 2871 : FIND = 2556
7  VIOLET = 85 : WHITE = 127
10 XO = 140 : YO = 80 : COLR = WHITE :
    SHAPE = 3 : ROT = 0 : SCALE = 1 : CALL DRAW
20 CALL FIND
30 XO = 10 : YO = 25 : COLR = VIOLET : CALL LINE
40 END

```

page 18

COLLISIONS

Overlapping shapes define points of 'collision'. The DRAW and XDRAW subroutines return a collision count in the absolute location \$32A (810 decimal). The collision count will be constant for a fixed shape, rotation, scale, and background, provided that no collisions with other shapes are detected. The difference between the 'standard' collision value and the encountered value (while drawing a shape) is a true collision indicator.

```

100 CALL DRAW
110 COUNT = PEEK (810)

```

APPENDING THE HI-RES PREFIX

The HI-RES PREFIX program may be permanently appended to any BASIC programs you write, making a 2-step LOAD unnecessary. If you have the APPLE-II RENUMBER/APPEND program then treat the user-written program as the one with greater line numbers (despite the fact that it begins with line 0) and the HI-RES PREFIX program as the one with smaller line numbers. If you don't have the RENUMBER/APPEND program then the APPEND may be done manually as follows:

1. > LOAD (user program)
2. > POKE 0, PEEK (76)
> POKE 1, PEEK (77)
> POKE 76, PEEK (202)
> POKE 77, PEEK (203)
(user program is now hidden)
3. > LOAD (HI-RES PREFIX program)
4. > POKE 76, PEEK (0)
> POKE 77, PEEK (1)
5. > SAVE (combined program)

SUMMARY

<u>Subroutine</u>	<u>Calling address</u>	<u>Parameters</u>
INIT	2048	
CLEAR	2062	
BKGND	2865	COLR
POSN	2809	XO, YO, COLR
PLOT	2830	XO, YO, COLR
LINE	2836	XO, YO, COLR
DRAW	2871	XO, YO, COLR, SHAPE, ROT, SCALE
DRAW1	2874	SHAPE, ROT, SCALE
XDRAW	2884	XO, YO, COLR, SHAPE, ROT, SCALE
XDRAW1	2887	SHAPE, ROT, SCALE
FIND	2556	
SHAPE .LOAD	3001	

For NO TEXT display ----- >POKE -16302,0

For mixed GRAPHICS/TEXT ----- >POKE -16301,0

Select secondary screen display ----- >POKE -16299,0

Select primary screen display ----- >POKE -16300,0

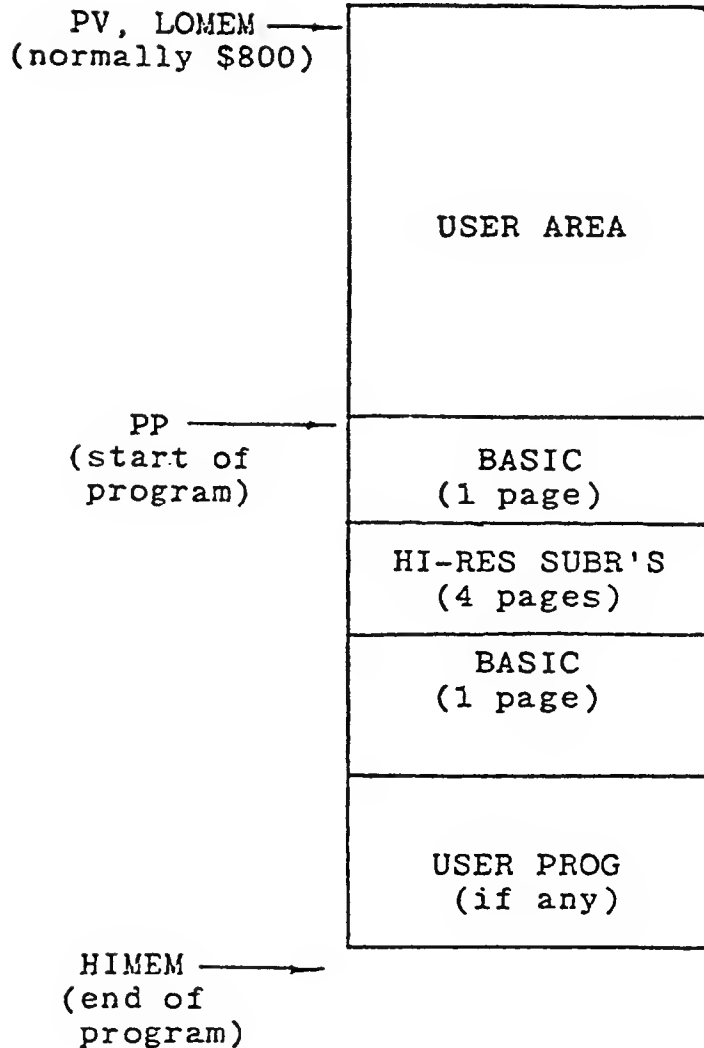
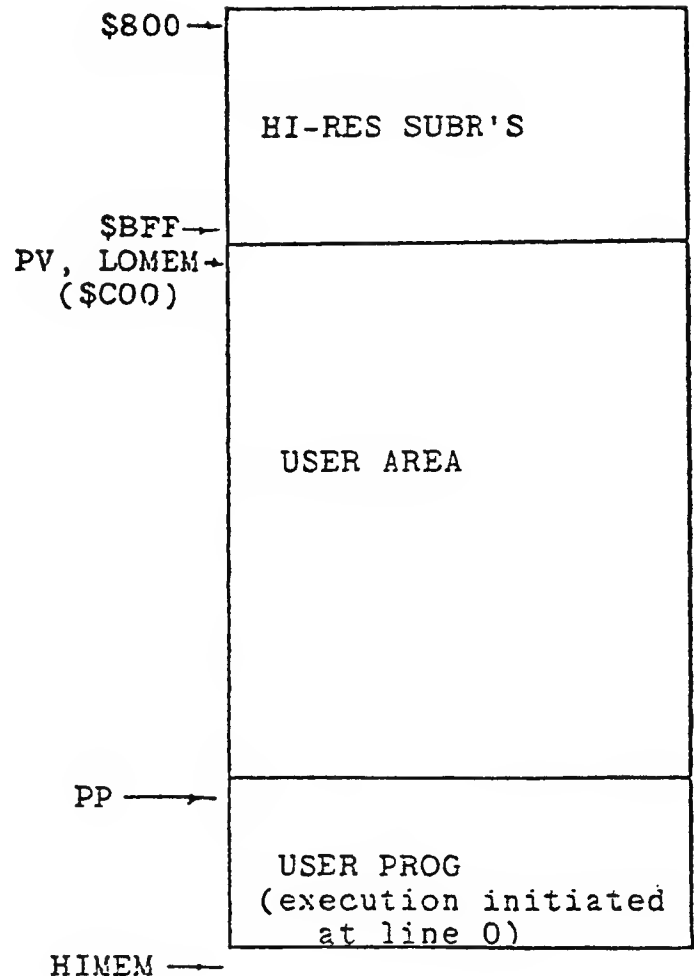
Select secondary screen plotting ----- >POKE 806,64

Select primary screen plotting ----- >POKE 806,32

(Defaults are GRAPHICS/TEXT, primary screen display,
and primary screen plotting)

Collision detect (shape draw only) ----- PEEK (S10)

HI-RES PREFIX LOAD

AFTER LOADAFTER RUN

Note: A 'page' is 256 bytes.

APPLE-II BASIC POINTERS

LOMEM (in \$4A, \$4B)-----Contains 'start of BASIC variables' address.

PV (\$CC, \$CD)-----End of BASIC variables. Equal to LOMEM if no active variables.

PP (\$CA, \$CB)-----Start of BASIC program. Equal to HIMEM if no program.

HIMEM (\$4C, \$4D)-----End of BASIC program.

HI-RES PARAMETER LOCATIONS (beyond LOMEM)

<u>Parameter</u>	<u>Locations beyond LOMEM</u>
X0	\$05, \$06
Y0	\$0C, \$0D
COLR	\$15, \$16
SHAPE	\$1F, \$20
ROT	\$27, \$28
SCALE	\$31, \$32

Note: Each parameter is two bytes in length. The low-order byte is stored in the lesser of the two locations assigned.

HI-RES SUBROUTINES SEGMENT MAP

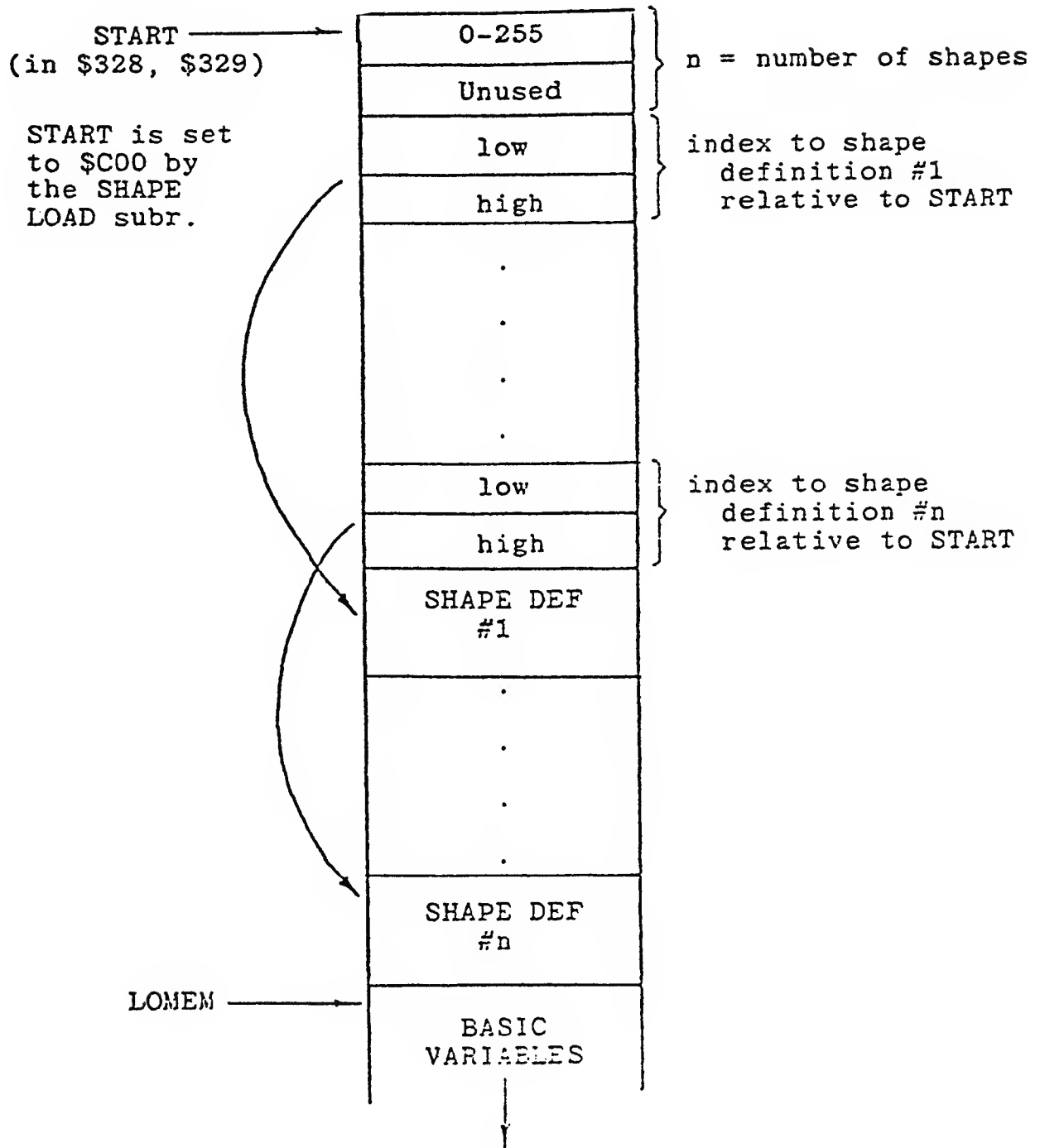
CODE \$800-\$9E8
DATA \$9E9-\$9FB
CODE \$9FC-\$BFF

SHAPE TAPE

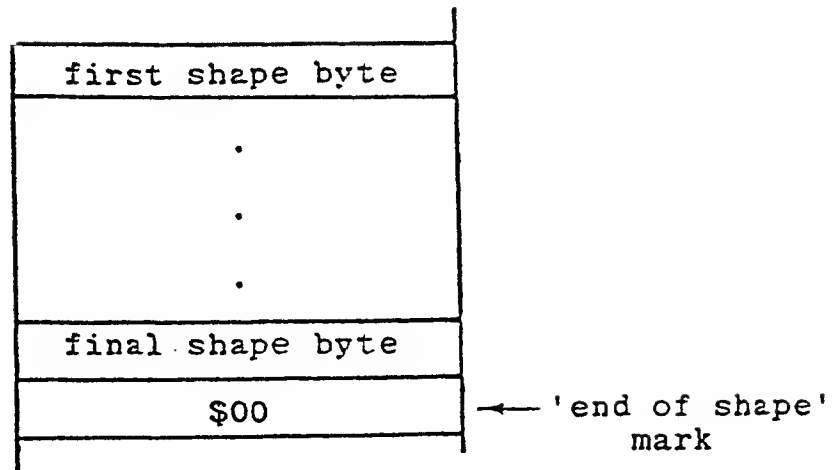
Record #1 ----- Contains length of record #2. Two bytes long,
low-order first.

Record Gap ----- Minimum of .7 seconds.

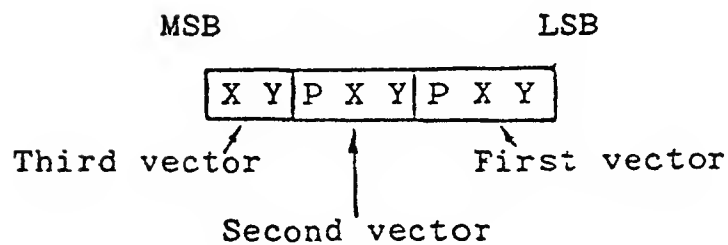
Record #2 ----- Shape table (see below).



SHAPE DEFINITIONS



SHAPE BYTE

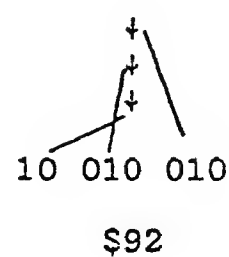
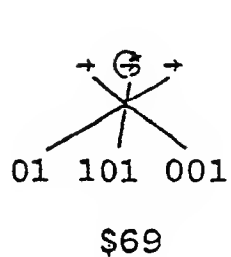
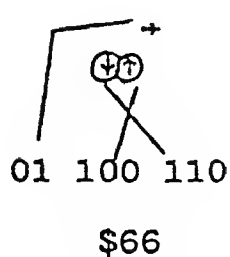
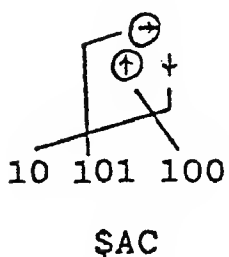
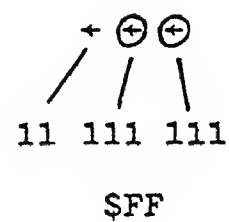
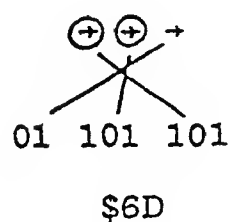
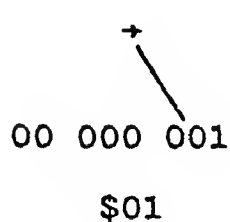
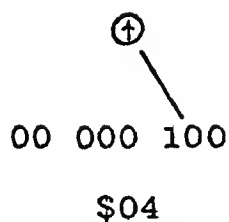
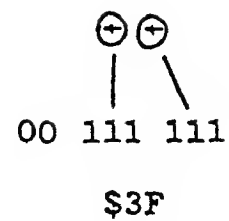
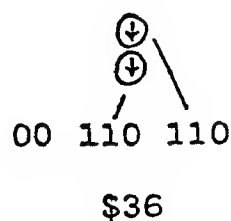
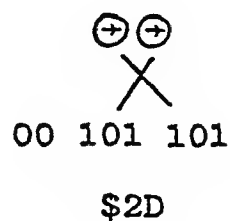
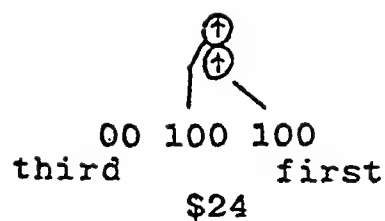


<u>X</u>	<u>Y</u>	<u>Vector</u>	
0	0	↑	
0	1	→	P = 0 Move without plot
1	0	↓	P = 1 Plot, then move
1	1	↖	Third vector is move without plo

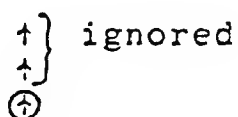
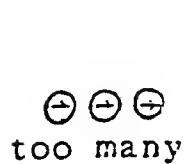
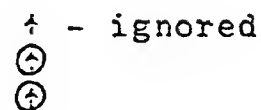
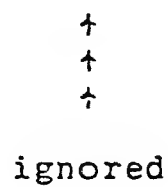
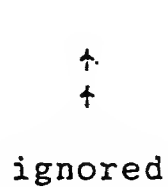
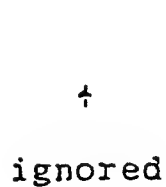
ZEROES ARE IGNORED--If the remaining one or two vectors of a shape byte are zeroes then they are ignored.

SAMPLE SHAPE BYTES

(plot-prior-move vectors are circled)



INVALID SHAPE BYTES



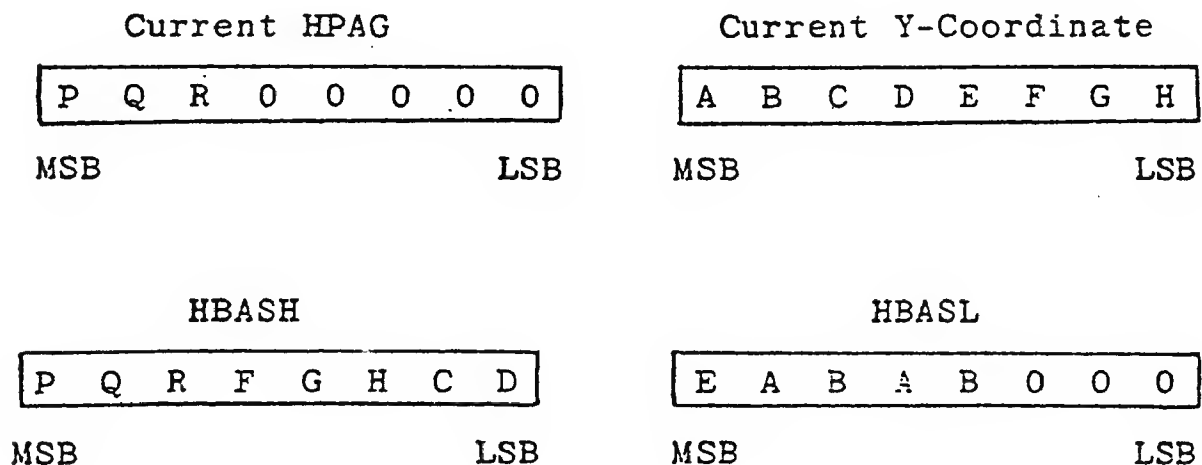
SCREEN MEMORY

1. HPAG (in location \$326) contains the high-order byte of the starting address of the current HI-RES display memory in which plotting is being done.

Primary screen memory plotting ----- HPAG = \$20
(\$2000-\$3FFF)

Secondary screen memory plotting ---- HPAG = \$40
(\$4000-\$5FFF)

2. HBASL and HBASH (in locations \$26 and \$27) contain the BASE ADDRESS corresponding to the current Y-coordinate. The BASE ADDRESS is the address of the leftmost display byte of the current line. HBASL and HBASH will track all plotting and drawing 'on-the-fly'.



3. HNDX (in location \$325) contains the byte index from the BASE ADDRESS to the current plot byte and is a function of the current X-coordinate.

$$\text{HNDX} = \text{X} / 7 \quad (\text{integer divide with truncate})$$

4. HMASK (in location \$30) contains a bit mask corresponding to the current bit position within the current plot byte and is a function of the current X-coordinate. The high-order bit is always set.

<u>X MOD 7</u>	<u>HMASK</u>
0 (leftmost)	\$81
1	\$82
2	\$84
3	\$88
4	\$90
5	\$A0
6 (rightmost)	\$C0

5. HCOLOR (in location \$1C) is the HI-RES 'on-the-fly' color mask. The low-order seven bits are rotated one bit position for odd values of HNDX. The high-order bit selects one of two color sets on systems modified for extra HI-RES colors.

<u>COLOR</u>	<u>HCOLOR</u>	
	<u>EVEN HNDX</u>	<u>ODD HNDX</u>
BLACK	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
GREEN	0 0 1 0 1 0 1 0	0 1 0 1 0 1 0 1
VIOLET	0 1 0 1 0 1 0 1	0 0 1 0 1 0 1 0
WHITE	0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1
BLACK2	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
ORANGE	1 0 1 0 1 0 1 0	1 1 0 1 0 1 0 1
BLUE	1 1 0 1 0 1 0 1	1 0 1 0 1 0 1 0
WHITE2	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

HI-RES INTERNAL VARIABLES

SHAPEL, SHAPEH (\$1A, \$1B) On-the-fly shape pointer.

HCOLOR1 (\$1C) On-the-fly color byte.

COUNTH (\$1D) High-order byte of step count for LINE.

HBASL, HBASH (\$26, \$27) On-the-fly BASE ADDRESS

HMASK (\$30) On-the-fly BIT MASK.

QDRNT (\$53) 2 LSB's are rotation quadrant for DRAW.

XOL, XOH (\$320, \$321) Most recent X-coordinate. Used for
initial endpoint of LINE. Updated
by PLOT, LINE, and FIND, not DRAW.

YO (\$322) Most recent Y-coordinate (see XOL, XOH).

BXSAV (\$323) Saves 6502 X-Register during HI-RES calls
from BASIC.

HCOLOR (\$324) Color specification to PLOT, POSN.

HNDX (\$325) On-the-fly byte index from BASE ADDRESS.

HPAG (\$326) Starting page of plot memory. Normally
\$20 for plotting in primary HI-RES
display memory (\$2000-\$3FFF).

SCALE (\$327) On-the-fly scale factor for DRAW.

SHAPXL, SHAPXH (\$328, \$329) Start of shape table pointer.

COLLSN (\$32A) Collision count from DRAW, XDRAW.

4:50. P.M., 12/2/1977

HI-RES EQUATES

PAGE 2

14	SHAPEL	EPZ	\$1A	POINTER TO
15	SHAPEH	EPZ	\$1B	SHAPE LIST
16	HCOLORI	EPZ	\$1C	RUNNING COLOR MASK.
17	COUNTH	EPZ	\$1D	
18	HBASL	EPZ	\$26	BASE ADR FOR CURRENT
19	HBASH	EPZ	\$27	HI-RES PLOT LINE.
20	HMASK	EPZ	\$30	
21	AIL	EPZ	\$3C	MONITOR A1.
22	A1H	EPZ	\$3D	
23	A2L	EPZ	\$3E	MONITOR A2.
24	A2H	EPZ	\$3F	
25	LOHEML	EPZ	\$4A	BASIC 'START OF VARS'
26	LOMEMH	EPZ	\$4B	
27	DXL	EPZ	\$50	DELTA-X FOR HLIN, SHAP
28	DXH	EPZ	\$51	
29	SHAPEX	EPZ	\$51	SHAPE TEMP.
30	DY	EPZ	\$52	DELTA-Y FOR HLIN, SHAP
31	QDRNT	EPZ	\$53	ROT QUADRANT (SHAPE).
32	EL	EPZ	\$54	ERROR FOR HLIN.
33	EH	EPZ	\$55	
34	PPL	EPZ	\$CA	BASIC START OF PROG. PR
35	PPH	EPZ	\$CB	
36	PVL	EPZ	\$CC	BASIC END OF VARS PTR.
37	PVH	EPZ	\$CD	
38	ACL	EPZ	\$CE	BASIC ACC.
39	ACH	EPZ	\$CF	
40	XOL	EQU	\$320	PRIOR X-COORD SAVE
41	XOH	EQU	\$321	AFTER HLIN OR HPLOT.
42	YO	EQU	\$322	HLIN, HPLOT Y-COORD SAV
43	BXSAV	EQU	\$323	X-REG SAVE FOR BASIC.
44	HCOLOR	EQU	\$324	COLOR FOR HPLOT, HPOSN
45	HNDX	EQU	\$325	HORIZ OFFSET SAVE.
46	HPAG	EQU	\$326	HI-RES PAGE (\$20 NORMA
47	SCALE	EQU	\$327	SCALE FOR SHAPE, MOVE.
48	SHAPXL	EQU	\$328	START OF
49	SHAPXH	EQU	\$329	SHAPE TABLE.
50	COLLSN	EQU	\$32A	COLLISION COUNT.
51	SHSTRT	EQU	\$C00	START OF SHAPE TABLE.
52	HIRES	EQU	\$C057	SWITCH TO HI-RES VIDEO
53	MIXSET	EQU	\$C053	SELECT TEXT/GRAPHICS
54	TXTCLE	EQU	\$C050	SELECT GRAPHICS MODE.
55	MEMFULL	EQU	\$E368	BASIC MEM FULL ERROR.
56	RANGERR	EQU	\$EE68	BASIC RANGE ERROR.
57	ACADR	EQU	\$F11E	2-BYTE TAPE READ SETUP
58	RD2BIT	EQU	\$FCFA	TWO-EDGE TAPE SENSE.
59	READ	EQU	\$FEFD	TAPE READ (A1.A2).
60	READX1	EQU	\$FF02	READ WITHOUT HEADER.

		63	*			
		64	*	RAM VERSION \$800 TO \$BFF		
		65	*			
		66		ORG \$800		
0800:	A9 20	67	SETHRL	LDA #S20		INIT FOR \$2000-3FFF
0802:	8D 26 03	68		STA HPAG		HI-RES SCREEN MEMORY.
0805:	AD 57 C0	69		LDA HIRES		SET HIRES DISPLAY MODE
0808:	AD 53 C0	70		LDA MIXSET		WITH TEXT AT BOTTOM.
080B:	AD 50 C0	71		LDA TXTCLR		SET GRAPHICS DISPLAY
080E:	A9 00	72	HCLR	LDA #S0		
0810:	85 1C	73	BKGND0	STA HCOLOR1		SET FOR BLACK BKGND.
0812:	AD 26 03	74	BKGND	LDA HPAG		
0815:	85 1B	75		STA SHAPEH		INIT HI-RES SCREEN MEM
0817:	A0 00	76		LDY #S0		FOR CURRENT PAGE, NORM
0819:	84 1A	77		STY SHAPEL		\$2000-3FFF OR \$4000-5F
081B:	A5 1C	78	BKGND1	LDA HCOLOR1		
081D:	91 1A	79		STA (SHAPEL),Y		
081F:	20 A2 08	80		JSR CSHFT2		(SHAPEL,H) WILL SPECIF
0822:	C8	81		INY		32 SEPARATE PAGES
0823:	D0 F6	82		BNE BKGND1		THROUGHOUT THE INIT.
0825:	E6 1B	83		INC SHAPEH		
0827:	A5 1B	84		LDA SHAPEH		
0829:	29 1F	85		AND #S1F		TEST FOR DONE.
082B:	D0 EE	86		BNE BKGND1		
082D:	60	87		RTS		

HI-RES GRAPHICS POSITION AND PLOT SUBRS

4:50 P.M., 12/2/1977

PAGE: 4

062E:	8D 22 03 90	HPOSN	STA	Y0	ENTER WITH Y IN A-REG,
0631:	8E 20 03 91		STX	X0L	XL IN X-REG,
0634:	8C 21 03 92		STY	X0H	AND XH IN Y-REG.
0637:	48		PHA		
0638:	29 C0		AND	#5C0	
063A:	85 26		STA	HBASL	FOR Y-COORD = 00ABCDEF
063C:	4A		LSR	A	CALCULATES BASE ADDR
063D:	4A		LSR	A	IN HBASL, HBASH FOR
063E:	05 26		ORA	HBASL	ACCESSING SCREEN MEM
0640:	85 26		STA	HBASL	VIA (HBASL),Y ADDR
0642:	68		PLA		MODE.
0643:	85 27		STA	HBASH	
0645:	0A		ASL	A	CALCULATES
0646:	0A		ASL	A	HBASH = PPPFGHCD,
0647:	0A		ASL	A	HBASL = EABAB000
0648:	26 27		ROL	HBASH	
064A:	0A		ASL	A	WHERE PPP=001 FOR \$20B
064B:	26 27		ROL	HBASH	SCREEN MEM RANGE AND
064D:	0A		ASL	A	PPP=010 FOR \$4000-7E
064E:	66 26		ROR	HBASL	(GIVEN Y-COORD=ABCDEF)
0650:	A5 27		LDA	HBASH	
0652:	29 1F		AND	#51F	
0654:	0D 26 03 112		ORA	HPAG	
0657:	85 27		STA	HBASH	
0659:	8A		TXA		DIVIDE X0 BY 7 FOR
065A:	C0 00		CPY	#50	INDEX FROM BASE ADR
065C:	F0 05		BEQ	HPOSN2	(QUOTIENT) AND BIT
065E:	A0 23		LDY	#523	WITHIN SCREEN MEM BE
0660:	69 04		ADC	#54	(MASK SPEC'D BY REME
0662:	C8		INY		
0663:	E9 07	HPOSN1	SBC	#57	SUBTRACT OUT SEVENS.
0665:	B0 FB	HPOSN2	BCS	HPOSN1	
0667:	8C 25 03 122		STY	HNDX	WORKS FOR X0 FROM
066A:	AA		TAX		0 TO 279, LOW-ORDER
066B:	BD EA 08 124		LDA	MSKTBL-\$F9,X	BYTE IN X-REG,
066E:	85 30		STA	HMASK	HIGH IN Y-REG ON EN
0670:	98		TYA		
0671:	4A		LSR	A	IF ON ODD BYTE (CARRY
0672:	AD 24 03 128		LDA	HCOLOR	THEN ROTATE HCOLOR 5
0675:	85 1C	HPOSN3	STA	HCOLOR1	BIT FOR 180 DEGREE R
0677:	B0 29		BCS	CSHFT2	PRIOR TO COPYING TOP
0679:	60		RTS		
067A:	20 2E 08 132	HPL0T	JSR	HPOSN	
067D:	A5 1C	HPL0T1	LDA	HCOLOR1	CALC BIT POSN IN HBASE
067F:	51 26		EOR	(HBASL),Y	HNDX, AND HMASK FROM
0681:	25 30		AND	HMASK	Y-COORD IN A-REG,
0683:	51 26		EOR	(HBASL),Y	X-COORD IN X,Y-REGS.
0685:	91 26		STA	(HBASL),Y	FOR ANY '1' BITS OF H
0687:	60		RTS		SUBSTITUTE CORRESPON
	139	*			BIT OF HCOLOR1.

4:50 P.M., 12/2/1977

0888:	10 24	142	LFTRT	BPL	RIGHT	USE SIGN FOR LFT/RT SE
088A:	A5 30	143	LEFT	LDA	HMASK	
088C:	4A	144		LSR	A	SHIFT LOW-ORDER
088D:	B0 05	145		BCS	LEFT1	7 BITS OF HMASK
088F:	49 C0	146		EOR	#5C0	ONE BIT TO LSB.
391:	85 30	147	LRI	STA	HMASK	
0893:	60	148		RTS		
0894:	58	149	LEFT1	DEY		DECR HORIZ INDEX.
0895:	10 02	150		BPL	LEFT2	
0897:	A0 27	151		LDY	#527	WRAP AROUND SCREEN.
0899:	A9 C0	152	LEFT2	LDA	#5C0	NEW HMASK, RIGHTMOST
089B:	85 30	153	NEWNDX	STA	HMASK	DOT OF BYTE.
089D:	8C 25 03	154		STY	HNDX	UPDATE HORIZ INDEX.
08A0:	A5 1C	155	CSHIFT	LDA	HCOLOR1	
08A2:	0A	156	CSHFT2	ASL	A	ROTATE LOW-ORDER
08A3:	C9 C0	157		CMP	#5CQ	7 BITS OF HCOLOR1
08A5:	10 06	158		BPL	RTS1	ONE BIT POSN.
08A7:	A5 1C	159		LDA	HCOLOR1	
08A9:	49 7F	160		EOR	#57F	ZXYXYXYX -> ZYXYXYXY
08AB:	85 1C	161		STA	HCOLOR1	
08AD:	60	162	RTS1	RTS		
08AE:	A5 30	163	RIGHT	LDA	HMASK	
08B0:	0A	164		ASL	A	SHIFT LOW-ORDER
08B1:	49 80	165		EOR	#580	7 BITS OF HMASK
08B3:	30 DC	166		BMI	LRI	ONE BIT TO MSB.
08B5:	A9 81	167		LDA	#581	
08B7:	C8	168		INY		NEXT BYTE.
08B8:	C0 28	169		CPY	#528	
08BA:	90 DF	170		BCC	NEWNDX	
8BC:	A0 00	171		LDY	#50	WRAP AROUND SCREEN IF
08BE:	B0 DB	172		BCS	NEWNDX	ALWAYS TAKEN.

4:50 P.M., 12/2/1977

PAGE: 6

08C0:	18	175	LRUDX1	CLC		NO 90 DEG ROT (X-OR).
08C1:	A5 51	176	LRUDX2	LDA	SHAPEX	
08C3:	29 04	177		AND	#54	IF B2=0 THEN NO PLOT.
08C5:	F0 27	178		BEQ	LRUD4	
08C7:	A9 7F	179		LDA	#57F	FOR EX-OR INTO SCREEN
08C9:	25 30	180		AND	HMASK	
08CB:	31 26	181		AND	(HBASL),Y	SCREEN BIT SET?
08CD:	D0 1B	182		BNE	LRUD3	
08CF:	EE 2A 03	183		INC	COLLSN	
08D2:	A9 7F	184		LDA	#57F	
08D4:	25 30	185		AND	HMASK	
08D6:	10 12	186		BPL	LRUD3	ALWAYS TAKEN.
08D8:	18	187	LRUD1	CLC		NO 90 DEG ROT.
08D9:	A5 51	188	LRUD2	LDA	SHAPEX	
08DB:	29 04	189		AND	#54	IF B2=0 THEN NO PLOT.
08DD:	F0 0F	190		BEQ	LRUD4	
08DF:	B1 26	191		LDA	(HBASL),Y	
08E1:	45 1C	192		EOR	HCOLOR1	SET HI-RES SCREEN BIT
08E3:	25 30	193		AND	HMASK	TO CORRESPONDING HCB
08E5:	D0 03	194		BNE	LRUD3	IF BIT OF SCREEN CHANS
08E7:	EE 2A 03	195		INC	COLLSN	THEN INCR COLLSN DE
08EA:	51 26	196	LRUD3	EOR	(HBASL),Y	
08EC:	91 26	197		STA	(HBASL),Y	
08EE:	A5 51	198	LRUD4	LDA	SHAPEX	ADD QDRNT TO
08F0:	65 53	199		ADC	QDRNT	SPECIFIED VECTOR
08F2:	29 03	200		AND	#53	AND MOVE LFT, RT,
		201	EQ3	EQU	*-1	UP, OR DWN BASED
08F4:	C9 02	202		CMP	#52	ON SIGN AND CARRY.
08F6:	6A	203		ROR	A	
08F7:	B0 8F	204	LRUD	BCS	LFTRT	
08F9:	30 30	205	UPDWN	BMI	DOWN4	SIGN FOR UP/DWN SELECT
08FB:	18	206	UP	CLC		
08FC:	A5 27	207		LDA	HBASH	CALC BASE ADDRESS
08FE:	2C EA 09	208		BIT	EQ1C	(ADR OF LEFTMOST BYT
0901:	D0 22	209		BNE	UP4	FOR NEXT LINE UP
0903:	06 26	210		ASL	HBASL	IN (HBASL,HBASH)
0905:	B0 1A	211		BCS	UP2 =	WITH 192-LINE WRAPAR
0907:	2C F3 08	212		BIT	EQ3	
090A:	F0 05	213		BEQ	UP1	
090G:	69 1F	214		ADC	#51F	**** BIT MAP ****
090E:	38	215		SEC		
090F:	B0 12	216		BCS	UP3	FOR ROW = ABCDEFGH,
0911:	69 23	217	UP1	ADC	#523	
0913:	48	218		PHA		
0914:	A5 26	219		LDA	HBASL	HBASL = EABAB000
0916:	69 B0	220		ADC	#5B0	HBASH = PPPFGHCD
0918:	B0 02	221		BCS	UP5	
091A:	69 F0	222		ADC	#5F0	WHERE PPP=001 FOR PRIE
091C:	85 26	223	UP5	STA	HBASL =	HI-RES PAGE (\$2000-F
091E:	68	224		PLA		
091F:	B0 02	225		BCS	UP3	
0921:	69 1F	226	UP2	ADC	#51F	
0923:	66 26	227	UP3	ROR	HBASL	
0925:	69 F0	228	UP4	ADC	#5FC	

4:50 P.M., 12/2/1977

LOAD SUBROUTINES

PAGE: 7

0927:	85	27	229	UPDOWN1	STA	HBASH
0929:	60		230		RTS	
092A:	18		231	DOWN	CLC	
092B:	A5	27	232	DOWN4	LDA	HBASH
092D:	69	04	233		ADC	#S4
			234	EQA	EQU	*-1
092F:	2C	EA	09 235		BIT	EQIC
0932:	D0	F3	236		BNE	UPDOWN1
0934:	06	26	237		ASL	HBASL
0936:	90	19	238		BCC	DOWN1
0938:	69	E0	239		ADC	#SE0
093A:	18		240		CLC	
093B:	2C	2E	09 241		BIT	EQ4
093E:	F0	13	242		BEQ	DOWN2
0940:	A5	26	243		LDA	HBASL
0942:	69	50	244		ADC	#S50
0944:	49	F0	245		EOR	#SF0
0946:	F0	02	246		BEQ	DOWN3
0948:	49	F0	247		EOR	#SF0
094A:	85	26	248	DOWN3	STA	HBASL
094C:	AD	26	03 249		LDA	HPAG
094F:	90	02	250		BCC	DOWN2
0951:	69	E0	251	DOWN1	ADC	#SE0
0953:	66	26	252	DOWN2	ROR	HBASL
0955:	90	D0	253		BCC	UPDOWN1

CALC BASE ADR FOR NEXT
DOWN TO (HBASL,HBASH

WITH 192-LINE WRAPAR

4:50 P.M., 12/2/1977

0957:	48	256	HLINRL	PHA		
0958:	A9 00	257		LDA	#50	SET (XOL,XOH) AND
095A:	8D 20 03	258		STA	XOL	YO TO ZERO FOR
095D:	8D 21 03	259		STA	XOH	REL LINE DRAW
960:	8D 22 03	260		STA	YO	(DX, DY).
0963:	68	261		PLA		
0964:	48	262	HLIN	PHA		ON ENTRY
0965:	38	263		SEC		XL: A-REG
0966:	ED 20 03	264		SBC	XOL	XH: X-REG
0969:	48	265		PHA		Y: Y-REG
096A:	8A	266		TXA		
096B:	ED 21 03	267		SBC	XOH	
096E:	85 53	268		STA	QDRNT	CALC ABS(X-XO)
0970:	B0 0A	269		BCS	HLIN2	IN (DXL,DXH)
0972:	68	270		PLA		
0973:	49 FF	271		EOR	#5FF	X DIR TO SIGN BIT
0975:	69 01	272		ADC	#51	OF QDRNT.
0977:	48	273		PHA		0=RIGHT (DX POS)
0978:	A9 00	274		LDA	#50	1=LEFT (DX NEG)
097A:	E5 53	275		SBC	QDRNT	
097C:	85 51	276	HLIN2	STA	DXH	
097E:	85 55	277		STA	EH	INIT (EL,EH) TO
0980:	68	278		PLA		ABS(X-XO)
0981:	85 50	279		STA	DXL	
0983:	85 54	280		STA	EL	
0985:	68	281		PLA		
0986:	8D 20 03	282		STA	XOL	
0989:	8E 21 03	283		STX	XOH	
98C:	98	284		TYA		
098D:	18	285		CLC		
098E:	ED 22 03	286		SBC	YO	CALC -ABS(Y-YO)-1
0991:	90 04	287		BCC	HLIN3	IN DY.
0993:	49 FF	288		EOR	#5FF	
0995:	69 FE	289		ADC	#5FE	
0997:	85 52	290	HLIN3	STA	DY	ROTATE Y DIR INTO
0999:	8C 22 03	291		STY	YO	QDRNT SIGN BIT
099C:	66 53	292		ROR	QDRNT	(0=UP, 1=DOWN)
099E:	38	293		SEC		
099F:	E5 50	294		SBC	DXL	INIT (COUNTL,COUNTH)
09A1:	AA	295		TAX		TO -(DELTX+DELT Y+1)
09A2:	A9 FF	296		LDA	#5FF	
09A4:	E5 51	297		SBC	DXH	
09A6:	85 1D	298		STA	COUNTH	
09A5:	AC 25 03	299		LDY	HNDX	HORIZ INDEX
09AB:	B0 05	300		BCS	MOVEX2	ALWAYS TAKEN.
09AD:	0A	301	MOVEX	ASL	A	MOVE IN X-DIR. USE
09AE:	20 88 08	302		JSR	LFTRT	QDRNT B6 FOR LFT/RTG
09B1:	38	303		SEC		
09B2:	A5 54	304	MOVEX2	LDA	EL	ASSUME CARRY SET.
09B4:	65 52	305		ADC	DY	(EL,EH)-DELT Y TO (EL,E
09B6:	85 54	306		STA	EL	NOTE: DY IS (-DELT Y)-1
09B8:	A5 55	307		LDA	EH	CARRY CLR IF (EL,EH).
09BA:	E9 00	308		SBC	#50	GOES NEG.
09BC:	85 55	309	HCOUNT	STA	EH	

HI-RES GRAPHICS LINE DRAW SUBRS

PAGE: 9

4:50 P.M., 12/2/1977

09E2:	B1 26	310	LDA	(HBASL),Y	SCREEN BYTE.
09C0:	45 1C	311	EOR	HCOLOR1	PLOT DOT OF HCOLOR1
09C2:	25 30	312	AND	HMASK	CURRENT BIT MASK.
09C4:	51 26	313	EOR	(HBASL),Y	
09C6:	91 26	314	STA	(HBASL),Y	
09C8:	E8	315	INX		DONE (DELTX+DELT Y)
09C9:	D0 04	316	BNE	HLIN4	DOTS?
09CB:	E6 1D	317	INC	COUNTH	
09CD:	F0 6B	318	BEQ	RTS2	YES, RETURN.
09CF:	A5 53	319	LDA	QDRNT	FOR DIRECTION TEST.
09D1:	B0 DA	320	BCS	MOVEX	IF CAR SET, (EL,EH) PG
09D3:	20 F9 08	321	JSR	UPDWN	IF CLR, NEG, MOVE YR
09D6:	18	322	CLC		
09D7:	A5 54	323	LDA	EL	(EL,EH)+DELTX
09D9:	65 50	324	ADC	DXL	TO (EL,EH).
09DB:	85 54	325	STA	EL	
09DD:	A5 55	326	LDA	EH	CAR SET IF (EL,EH) GO
09DF:	65 51	327	ADC	DXH	
09E1:	50 D9	328	BVC	HCOUNT	ALWAYS TAKEN.
09E3:	81	329	DBT	\$81	LEFTMOST BIT OF BYTE.
09E4:	82 84 88	330	DBT	\$82,\$84,\$88	
09E7:	90 A0	331	DBT	\$90,\$A0	
09E9:	C0	332	DBT	\$C0	RIGHTMOST BIT OF BYTE.
09EA:	1C	333	DBT	\$1C	
09EB:	FF FE FA				
09EE:	F4 EC E1				
09F1:	D4 C5 B4	334	DBT	\$FF,\$FE,\$FA,\$	
09F4:	A1 8D 78				
09F7:	61 49 31				
09FA:	18 FF	335	DBT	\$A1,\$8D,\$78,\$	

4:50 P.M., 12/2/1977

09FC:	A5 26	338	HFIND	LDA	HBASL	
09FE:	0A	339		ASL	A	CONVERTS BASE ADR
09FF:	A5 27	340		LDA	HBASH	TO Y-COORD.
0A01:	29 03	341		AND	#53	
0A03:	2A	342		ROL	A	FOR HBASL = EABAB000
0A04:	05 26	343		ORA	HBASL	HBASH = PPPFGHCD
0A06:	0A	344		ASL	A	
0A07:	0A	345		ASL	A	GENERATE
0A08:	0A	346		ASL	A	Y-COORD = ABCDEFGH
0A09:	8D 22 03	347		STA	Y0	
0A0C:	A5 27	348		LDA	HBASH	(PPP=SCREEN PAGE,
0A0E:	4A	349		LSR	A	NORMALLY 001 FOR
0A0F:	4A	350		LSR	A	\$2000-\$3FFF
0A10:	29 07	351		AND	#57	HI-RES SCREEN)
0A12:	0D 22 03	352		ORA	Y0	
0A15:	8D 22 03	353		STA	Y0	CONVERTS HNDX (INDEX
0A18:	AD 25 03	354		LDA	HNDX	FROM BASE ADR)
0A1B:	0A	355		ASL	A	AND HMASK (BIT
0A1C:	6D 25 03	356		ADC	HNDX	MASK) TO X-COORD
0A1F:	0A	357		ASL	A	IN (XOL,XOH)
0A20:	AA	358		TAX		(RANGE \$0-\$133)
0A21:	CA	359		DEX		
0A22:	A5 30	360		LDA	HMASK	
0A24:	29 7F	361		AND	#57F	
0A26:	E8	362	HFIND1	INX		
0A27:	4A	363		LSR	A	
0A28:	D0 FC	364		BNE	HFIND1	
0A2A:	8D 21 03	365		STA	XOH	
0A2D:	8A	366		TXA		
0A2E:	18	367		GLC		CALC HNDX*7 +
0A2F:	6D 25 03	368		ADC	HNDX	LOG (BASE 2) HMASK.
0A32:	90 03	369		BCC	HFIND2	
0A34:	EE 21 03	370		INC	XOH	
0A37:	8D 20 03	371	HFIND2	STA	XOL	
0A3A:	60	372	RTS2	RTS		

4:50 P.M., 12/2/1977

```

375 *
376 * SHAPE DRAW
377 * R = 0 TO 63
378 * SCALE FACTOR USED (1=NORMAL)
379 *
0A3B: 86 1A 380 DRAW STX SHAPEL DRAW DEFINITION
0A3D: 84 1B 381 STY SHAPEH POINTER.
0A3F: AA 382 DRAW1 TAX
0A40: 4A 383 LSR A ROT (S0-S3F)
0A41: 4A 384 LSR A
0A42: 4A 385 LSR A QDRNT 0=UP, 1=RT,
0A43: 4A 386 LSR A 2=DWN, 3=LFT.
0A44: 85 53 387 STA QDRNT
0A46: 8A 388 TXA
0A47: 29 0F 389 AND #SF
0A49: AA 390 TAX
0A4A: BC EB 09 391 LDY COS,X SAVE COS AND SIN
0A4D: 84 50 392 STY DXL VALS IN DXL AND DY.
0A4F: 49 0F 393 EOR #SF
0A51: AA 394 TAX
0A52: BC EC 09 395 LDY COS+1,X
0A55: C8 396 INY
0A56: 84 52 397 STY DY
0A58: AC 25 03 398 DRAW2 LDY HNDX BYTE INDEX FROM
0A5B: A2 00 399 LDX #S0 HI-RES BASE ADR.
0A5D: 8E 2A 03 400 STX COLLN CLEAR COLLISION COUNT.
0A60: A1 1A 401 LDA (SHAPEL,X) 1ST SHAPE DEF BYTE.
0A62: 85 51 402 DRAW3 STA SHAPEX
0A64: A2 80 403 LDX #S80
0A66: 86 54 404 STX EL EL,EH FOR FRACTIONAL
0A68: 86 55 405 STX EH L,R,U,D VECTORS.
0A6A: AE 27 03 406 LDX SCALE SCALE FACTOR.
0A6D: A5 54 407 DRAW4 LDA EL
0A6F: 38 408 SEC IF FRAC COS OVFL
0A70: 65 50 409 ADC DXL THEN MOVE IN
0A72: 85 54 410 STA EL SPECIFIED VECTOR
0A74: 90 04 411 BCC DRAW5 DIRECTION.
0A76: 20 D8 08 412 JSR LRUD1
0A79: 18 413 CLC
0A7A: A5 55 414 DRAW5 LDA EH IF FRAC SIN OVFL
0A7C: 65 52 415 ADC DY THEN MOVE IN
0A7E: 85 55 416 STA EH SPECIFIED VECTOR
0A80: 90 03 417 BCC DRAW6 DIRECTION +90 DEG.
0A82: 20 D9 08 418 JSR LRUD2
0A85: CA 419 DRAW6 DEX LOOP ON SCALE
0A86: D0 E5 420 BNE DRAW4 FACTOR.
0A88: A5 51 421 LDA SHAPEX
0A8A: 4A 422 LSR A NEXT 3-BIT VECTOR
0A8B: 4A 423 LSR A OF SHAPE DEF.
0A8C: 4A 424 LSR A
0A8D: D0 D3 425 BNE DRAW3 NOT DONE THIS BYTE.
0A8F: E6 1A 426 INC SHAPEL
0A91: D0 02 427 BNE DRAW7 NEXT BYTE OF
0A93: E6 1B 428 INC SHAPEH SHAPE DEFINITION.

```

4:50 P.M., 12/2/1977

```

0A95: A1 1A 429 DRAW7 LDA (SHAPEL,X)
0A97: D0 C9 430 BNE DRAW3 DONE IF ZERO.
0A99: 60 431 RTS

```

		434	*		
		435	* EX-OR SHAPE INTO SCREEN.		
		436	*		
		437	* ROT = 0 TO 3 (QUADRANT ONLY)		
		438	* SCALE IS USED		
		439	*		
0A9A:	86 1A	440	XDRAW	STX SHAPEL	SHAPE DEFINITION
0A9C:	84 1B	441		STY SHAPEH	POINTER.
0A9E:	AA	442	XDRAW1	TAX	
0A9F:	4A	443		LSR A	ROT (50-53F)
0AA0:	4A	444		LSR A	
0AA1:	4A	445		LSR A	QDRNT 0=UP, 1=RT,
0AA2:	4A	446		LSR A	2=DWN, 3=LFT.
0AA3:	85 53	447		STA QDRNT	
0AA5:	8A	448		TXA	
0AA6:	29 0F	449		AND #5F	
0AA8:	AA	450		TAX	
0AA9:	BC EB 09	451		LDY COS,X	SAVE COS AND SIN
0AAC:	84 50	452		STY DXL	VALS IN DXL AND DY.
0AAZ:	49 0F	453		EOR #5F	
0ABQ:	AA	454		TAX	
0AB1:	BC EC 09	455		LDY COS+1,X	
0AB4:	C8	456		INY	
0AB5:	84 52	457		STY DY	
0AB7:	AC 25 03	458	XDRAW2	LDY HNDX	INDEX FROM HI-RES
0ABA:	A2 00	459		LDX #50	BASE ADR.
0ABC:	8E 2A 03	460		STX COLLN	CLEAR COLLISION DETECT
0ABF:	A1 1A	461		LDA (SHAPEL,X)	1ST SHAPE DEF BYTE.
0AC1:	85 51	462	XDRAW3	STA SHAPEX	
0AC3:	A2 80	463		LDX #580	
0AC5:	86 54	464		STX EL	EL,EH FOR FRACTIONAL
0AC7:	86 55	465		STX EH	L,R,U,D VECTORS.
0AC9:	AE 27 03	466		LDX SCALE	SCALE FACTOR.
0ACC:	A5 54	467	XDRAW4	LDA EL	
0ACE:	38	468		SEC	IF FRAC COS OVFL
0ACF:	65 50	469		ADC DXL	THEN MOVE IN
0AD1:	85 54	470		STA EL	SPECIFIED VECTOR
0AB3:	90 04	471		BCC XDRAWS	DIRECTION
0AD5:	20 C0 08	472		JSR LRJDX1	
0AD8:	18	473		CLC	
0AD9:	A5 55	474	XDRAW5	LDA EH	IF FRAC SIN OVFL
0ADB:	65 52	475		ADC DY	THEN MOVE IN
0ADD:	85 55	476		STA EH	SPECIFIED VECTOR
0ADF:	90 03	477		BCC XDRAW6	DIRECTION +90 DEG.
0AE1:	20 D9 08	478		JSR LRUD2	
0AE4:	CA	479	XDRAW6	DEX	LOOP ON SCALE
0AE5:	D0 E5	480		BNE XDRAW4	FACTOR.
0AE7:	A5 51	481		LDA SHAPEX	
0AE9:	4A	482		LSR A	NEXT 3-BIT VECTOR
0AEA:	4A	483		LSR A	OF SHAPE DEF.
0AEB:	4A	484		LSR A	
0AEC:	D0 D3	485		ENE XDRAW3	
0AEE:	E6 1A	486		INC SHAPEL	
0AF0:	D0 02	487		BNE XDRAW7	NEXT BYTE OF
0AF2:	E6 1B	488		INC SHAPEH	SHAPE DEF.
0AF4:	A1 1A	489	XDRAW7	LDA (SHAPEL,X)	
0AF6:	D0 C9	490		BNE XDRAW3	DONE IF ZERO.
0AF8:	60	491		RTS	

ENTRY POINTS FROM APPLE-II BASIC

PAGE: 15

4:50 P.M., 12/2/1977

0AF9:	20 90 0B 494	BPOSN	JSR PCOLR	POSN CALL, COLR FROM A
CAFC:	8D 24 03 495		STA HCOLOR	
0AFF:	20 AF 0B 496		JSR GETY0	Y0 FROM BASIC.
0B02:	48		PHA	
0B03:	20 9A 0B 498		JSR GETX0	X0 FROM BASIC.
0B06:	68		PLA	
0B07:	20 2E 08 500		JSR HPOSN	
0B0A:	AE 23 03 501		LDX BXSAY	
0B0D:	60		RTS	
0B0E:	20 F9 0A 503	BPLOT	JSR BPOSN	PLOT CALL (BASIC).
0B11:	4C 7D 0B 504		JMP HPL0T1	
0B14:	AD 25 03 505	BLINI	LDA HNDX	
0B17:	4A		LSR A	SET HCOLORI FROM
0B18:	20 90 0B 507		JSR PCOLR	BASIC VAR COLR.
0B1B:	20 75 08 508		JSR HPOSN3	
0B1E:	20 9A 0B 509	BLINE	JSR GETX0	LINE CALL, GET X0 FROM
0B21:	8A		TXA	
0B22:	48		PHA	
0B23:	98		TYA	
0B24:	AA		TAX	
0B25:	20 AF 0B 514		JSR GETY0	Y0 FROM BASIC
0B28:	A8		TAY	
0B29:	68		PLA	
0B2A:	20 64 09 517		JSR HLIN	
0B2D:	AE 23 03 518		LDX BXSAY	
0B30:	60		RTS	
0B31:	20 90 0B 520	BGND	JSR PCOLR	BACKGROUND CALL
0B34:	4C 10 0B 521		JMP BKGND0	

0B37:	20 F9 0A 524	BDRAW1	JSR	BPOSN	
0B3A:	20 51 0B 525	BDRW	JSR	BDRWXX	DRAW CALL FROM BASIC.
0B3D:	20 3B 0A 526		JSR	DRAW	
0B40:	AE 23 03 527		LDX	BXSAV	
0B43:	60		RTS		
44:	20 F9 0A 529	BXDRW1	JSR	BPOSN	
0B47:	20 51 0B 530	BXDRAW	JSR	BDRWXX	EX-OR DRAW
0B4A:	20 9A 0A 531		JSR	XDRAW	FROM BASIC.
0B4D:	AE 23 03 532		LDX	BXSAV	
0B50:	60		RTS		
0B51:	8E 23 03 534	BDRWXX	STX	BXSAV	SAVE FOR BASIC.
0B54:	A0 32		LDY	#532	
0B56:	20 92 0B 536		JSR	PBYTE	SCALE FROM BASIC.
0B59:	8D 27 03 537		STA	SCALE	
0B5C:	A0 28		LDY	#528	
0B5E:	20 92 0B 539		JSR	PBYTE	ROT FROM BASIC.
0B61:	48		PHA		SAVE ON STACK.
0B62:	AD 28 03 541		LDA	SHAPXL	
0B65:	85 1A		STA	SHAPEL	START OF
0B67:	AD 29 03 543		LDA	SHAPXH	SHAPE TABLE.
0B6A:	85 1B		STA	SHAPEH	
0B6C:	A0 20		LDY	#520	
0B6E:	20 92 0B 546		JSR	PBYTE	SHAPE FROM BASIC.
0B71:	F0 39		BEQ	RERR1	
0B73:	A2 00		LDX	#50	
0B75:	C1 1A		CMF	(SHAPEL,X)	> NUM OF SHAPES?
0B77:	F0 02		BEQ	BDRWX1	
0B79:	B0 31		BCS	RERR1	YES, RANGE ERR.
37B:	0A		ASL	A	
0B7C:	90 03	BDRWX1	BCC	BDRWX2	
0B7E:	E6 1B		INC	SHAPEH	
0B80:	18		CLC		
0B81:	A8	BDRWX2	TAY		SHAPE NO. * 2.
0B82:	B1 1A		LDA	(SHAPEL),Y	
0B84:	65 1A		ADC	SHAPEL	
0B86:	AA		TAX		ADD 2-BYTE INDEX
0B87:	C8		INY		TO SHAPE TABLE
0B88:	B1 1A		LDA	(SHAPEL),Y	START ADR
0B8A:	6D 29 03 562		ADC	SHAPXH	(X LOW, Y HI).
0B8D:	A8		TAY		
0B8E:	68		PLA		ROT FROM STACK.
0B8F:	60		RTS		

4:50 P.M., 12/2/1977

0B90:	A0 16	568	PCOLR	LDY	#S16	
0B92:	B1 4A	569	PBYTE	LDA	(LOMEML),Y	
0B94:	D0 16	570		BNE	RERR1	GET BASIC PARAM.
0B96:	B8	571		DEY		(ERR IF >255)
0B97:	B1 4A	572		LDA	(LOMEML),Y	
0B99:	60	573	RTSB	RTS		
0B9A:	8E 23 03	574	GETX0	STX	BXSAV	SAVE FOR BASIC.
0B9D:	AQ 05	575		LDY	#S5	
0B9F:	B1 4A	576		LDA	(LOMEML),Y	X0 LOW-ORDER BYTE.
0BA1:	AA	577		TAX		
0BA2:	C8	578		INY		
0BA3:	B1 4A	579		LDA	(LOMEML),Y	HI-ORDER BYTE.
0BA5:	A8	580		TAY		
0BA6:	E0 18	581		CPX	#S18	
0BA8:	E9 01	582		SBC	#S1	RANGE ERR IF >279.
0BAA:	90 ED	583		BCC	RTSB	
0BAC:	4C 68 EE	584	RERR1	JMP	RANGERR	
0BAF:	A0 0D	585	GETY0	LDY	#SD	OFFSET TO Y0 FROM LOMX
0BB1:	20 92 0B	586		JSR	PBYTE	GET BASIC PARAM Y0.
0BB4:	C9 C0	587		CMP	#SC0	(ERR IF >191)
0BB6:	B0 F4	588		BCS	RERR1	
0BB8:	60	589		RTS		

SHAPE TAPE LOAD SUBROUTINE

0 P.M., 12/2/1977

PAGE: 18

B9:	8E 23 03	592	SHLOAD	STX	BXSAV	SAVE FOR BASIC.
0B9C:	20 1E F1	593		JSR	ACADR	READ 2-BYTE LENGTH INTO
0B9F:	20 FD FE	594		JSR	READ	BASIC ACC (SCE,CF).
* WARNING: OPERAND OVERFLOW IN LINE 595						
0BC2:	A9 00	595		LDA	#SHSTRT	
0BC4:	85 3C	596		STA	A1L	
0BC6:	8D 28 03	597		STA	SHAPXL	
0BC9:	18	598		CLC		
0BCA:	65 CE	599		ADC	ACL	
0BCC:	A8	600		TAY		
0BCD:	A9 0C	601		LDA	#SHSTRT/256	
0BCF:	85 3D	602		STA	A1H	
0BD1:	8D 29 03	603		STA	SHAPXH	
0BD4:	65 CF	604		ADC	ACH	
0BD6:	B0 25	605		BCS	MFULL1	NOT ENOUGH MEMORY.
0BD8:	C4 CA	606		CPY	PPL	
0BDA:	48	607		PHA		
0BDB:	E5 CB	608		SBC	PPH	
0BDD:	68	609		PLA		
0BDE:	B0 1D	610		BCS	MFULL1	
0BE0:	84 3E	611		STY	A2L	
0BE2:	85 3F	612		STA	A2H	
0BE4:	C8	613		INY		
0BE5:	D0 02	614		BNE	SHLOD1	
0BE7:	69 01	615		ADC	#S1	
0BE9:	84 4A	616	SHLOD1	STY	LOMEML	
0BED:	85 4B	617		STA	LOMEMH	
0BED:	84 CC	618		STY	PVL	
0BEF:	85 CD	619		STA	PVH	
0BF1:	20 FA FC	620		JSR	RD2BIT	
0BF4:	A9 03	621		LDA	#S3	.5 SECOND HEADER.
0BF6:	20 02 FF	622		JSR	READX1	
0BF9:	AE 23 03	623		LDX	BXSAV	
0BFC:	60	624		RTS		